Jan Pelikán

Pickup and delivery problem with split demand and transfers

Kybernetika, Vol. 49 (2013), No. 5, 755--764

Persistent URL: http://dml.cz/dmlcz/143523

KYBERNETIKA — VOLUME *49* (2013), NUMBER 5, PAGES 755–764

PICKUP AND DELIVERY PROBLEM WITH SPLIT DEMAND AND TRANSFERS

Jan Pelikán

We deal with a logistic problem motivated by a case study from a company dealing with inland transportation of piece goods in regular cycles. The problem consists in transportation of goods among regional centres — hubs of a network. Demands on transportation are contained in a matrix of flows of goods between pairs of hubs. The transport is performed by vehicles covering the shipping demands and the task is to design a cyclical route and to place a depot for each vehicle. The route depot can be placed in any hub of the route. Goods can be transferred from one route and vehicle to another route and vehicle. The aim is to minimize the total transportation cost. The task is classified as a new case of the pickup and delivery problem with split demand and transfers (SDPDPT). We propose a mathematical model and prove NP-hardness of the problem. We study demand reducibility. We also deal with skip pickup and delivery problem as a special case and show its complexity.

Keywords: pickup and delivery problem, case study, integer programming, skip transportation

Classification: 90B35, 90B90

1. INTRODUCTION — DESCRIPTION OF THE CASE STUDY

The main problem is: how to organize inland transport of goods between pairs of logistical centres (hubs), where the hubs are placed on a network. The transportation is carried out by vehicles of the same capacity, each of which can be placed in any logistical centre, which is the depot for this vehicle. The hub is posed as node of the graph, arcs of the graph are the shortest paths between pairs of hubs. Demands on transport are formulated by a matrix $D = (d_{ij})$ representing the flows of goods between pairs of hubs. The number d_{ij} can be understood as the average daily demand on transport. The demand d_{ij} represents the amount of goods which will be loaded at node i and unloaded at node j. During the transportation process, the goods can be divided into parts, which can be transported independently on different vehicles, and can also be reloaded from one vehicle to another one in some of the vehicles' route nodes.

The task is to determine a set of cyclical routes of the graph and a depot on each cycle. Each route is assigned to one vehicle. The depot can be placed in any node on the route. The aim is to find a set of cyclical routes with minimum transportation cost. These routes can intersect themselves. In the intersections, the goods could be transferred from one vehicle to the other one if necessary. In practice, this concerns nine logistic centers in the Czech Republic.

2. PICKUP AND DELIVERY PROBLEM

There are problems in logistics where the goal is to assure pickup and delivery of goods in a distribution network. While in those applications it is necessary to find cyclical routes starting and ending at a given depot, in the studied problem it is required to transport goods between nodes of the network by cyclical routes with different depots. Each requirement is specified by the pickup node, the delivery node and the amount of goods which has to be transported. In Savelsbergh and Sol [5] the problem is called *pickup and delivery problem.* Definition of the pickup and delivery problem (Berbeglia et al. [2]) involves the general problem described by Savelsbergh and Sol [5] and also further logistic problems and their modifications: traveling salesman problem, vehicle routing problem, dial-a-ride problem, swapping problem, stacker crane problem. The dial-a-ride problem consists in covering a set of requests of people for transport from an origin to a destination, door to door transport service. The swapping problem deals with the swapping of objects among nodes of the network. The stacker crane problem is a practical problem of managing crane operations.

A tree-field classification scheme is presented by Berbeglia et al. [2] for the class of pickup and delivery problems. The first field structure is a specification of numbers of origins and destinations for transport of commodities. The second field provides information on pick and delivery operations at nodes. The third field gives the number of vehicles. The static and dynamic pickup and delivery problem with time windows, split deliveries and possible reloading is solved by Thangiah at al. [6]. Heuristics are proposed and verified on real data.

The problem studied in this paper is a new case of the pickup and delivery problem, in which demand can be split and reloaded. Route depot can be any node of the route.

3. MATHEMATICAL MODEL

The task introduced in the case study is classified as a new case of the pickup and delivery problem with split demand and transfers (SDPDPT). The problem is modeled with a complete digraph, where hubs are represented by nodes of the graph. Arcs represent the shortest paths between pairs of nodes. The number of nodes of the graph is denoted n. The lengths of arcs are given by a distance matrix $C = (c_{ij})$, where c_{ij} is the length of the shortest path from node i to node j in the road network. The demands on the flows of goods between pairs of nodes are given by a matrix $D = (d_{k\ell})$, where $d_{k\ell}$ represents the demand for transport from node k to node ℓ. The task is to determine a family $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ of cyclical routes $R_1, R_2, \ldots, R_m$ on the road graph. The length of the cyclical route R_j is denoted $h(R_j)$. The total length of routes is $\sum_j h(R_j)$ and this number is to be minimized.

All the vehicles have the same capacity $Q > 0$. The number of vehicles is not limited beforehand.

The mathematical model contains two types of flows in the graph:

- the flow of goods, which is a multiproduct flow in the graph; each demand on transport $d_{k\ell}$ corresponds to one product,
- the flow of vehicles. The capacity of an edge for goods flow is given by the number of vehicles going along this edge and the capacity of vehicle Q. The flow of vehicles along the edge is unlimited.

Mathematical model:

Parameters:

- C – matrix of the length of arcs,
- D – matrix of demands on transport between pairs of nodes,
- Q – capacity of the vehicle,
- n – number of nodes.

Variables:

- y_{ij} – number of vehicles going along the arc (i, j) $(i, j = 1, 2, \dots, n;\ i \neq j)$,
- $x_{ij}^{k\ell}$ – volume of goods transported from node i to node j, being a part of the total amount $d_{k\ell}$ of goods which are to be transported from node k to node ℓ $(i, j = 1, 2, \dots, n;\ i \neq j;\ k, \ell = 1, 2, \dots, n;\ k \neq \ell)$.

The SDPDPT model:

$$\min F(Y) = \sum_{i,j} c_{ij} y_{ij} \quad \text{subject to}$$

$$\sum_i y_{ij} - \sum_i y_{ji} = 0, \qquad j = 1, 2, \dots, n; \tag{1}$$

$$\sum_i x_{ij}^{k\ell} - \sum_i x_{ji}^{k\ell} = \left\{ \begin{array}{rl} -d_{k\ell}, & j = k \\ d_{k\ell}, & j = \ell \\ 0, & j \neq k, \ell \end{array} \right\}, \qquad k, \ell = 1, 2, \dots, n; \quad k \neq \ell; \tag{2}$$

$$\sum_{k\ell} x_{ij}^{k\ell} \leq Q y_{ij}, \qquad i, j = 1, 2, \dots, n; \quad i \neq j; \tag{3}$$

$$x_{ij}^{k\ell} \geq 0, \qquad k, \ell = 1, 2, \dots, n; \quad k \neq \ell; \quad i, j = 1, 2, \dots, n; \quad i \neq j; \tag{4}$$

$$y_{ij} \geq 0, \qquad y_{ij} \text{ integer}; \quad i, j = 1, 2, \dots, n; \quad i \neq j. \tag{5}$$

The objective (1) corresponds to the sum of costs of all arcs in the solution, i. e., the total length of all routes. Equations (2) are flow equations for vehicles: they say that the number of vehicles entering a node has to be equal to the number of vehicles leaving it. The equations (3) are flow equations for goods: they say that the amount of goods being transported from k to ℓ entering a node also leaves the node (except for the source and destination node). Inequalities (4) prevent the capacity of the vehicle transporting goods between nodes i and j from being exceeded.

Example 3.1. Consider a graph with nodes $V = \{1,2,3,4\}$, the demand matrix D and the cost matrix C of the form

$$D = \begin{pmatrix} 0 & 10 & 5 & 5 \\ 10 & 0 & 5 & 5 \\ 5 & 5 & 0 & 0 \\ 5 & 5 & 0 & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 3 & 2 & 2 \\ 3 & 0 & 2 & 2 \\ 2 & 2 & 0 & 3 \\ 2 & 2 & 3 & 0 \end{pmatrix}.$$

The capacity of a vehicle is $Q = 10$. The optimal solution of the model is

$$Y = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

with optimum cost 16.

Using the matrix Y, two cyclical routes can be found using an algorithm, which will be presented in the next section. The routes are

$$R_1 = (1,3,2,4,1), \quad R_2 = (1,4,2,3,1)$$

and the depot is in node 1.

A summary of loadings and unloadings of the goods is shown in Table 1. Demand d_{12} is divided into two parts: a part d'_{12}, which is transported along the cycle R_1, and a part d''_{12}, which is is transported along R_2. Clearly $d_{12} = d'_{12} + d''_{12}$. In a similar way, the demand d_{21} is divided into d'_{21} and d''_{21}.

R_1: nodes	Loaded	Unloaded	R_2: nodes	Loaded	Unloaded
1	$d'_{12} = 5, d_{13} = 5$	—	1	$d_{14} = 5, d''_{12} = 5$	—
3	$d_{32} = 5$	$d_{13} = 5$	4	$d_{42} = 5$	$d_{14} = 5$
2	$d_{24} = 5, d'_{21} = 5$	$d_{12} = 5, d_{32} = 5$	2	$d''_{21} = 5, d_{23} = 5$	$d''_{12} = 5, d_{42} = 5$
4	$d_{41} = 5$	$d_{24} = 5$	3	$d_{31} = 5$	$d_{23} = 5$
1	—	$d'_{21} = 5, d_{41} = 5$	1		$d_{31} = 5, d''_{21} = 5$

Tab. 1. Routes R_1 and R_2.

3.1. Generation of cyclical routes

A number of vehicles entering each node equals to the number of vehicles leaving it in the optimal solution Y of the model (1) – (6). When Y is available, we need to generate the family of the cyclical routes, each in the form of a path $(i_1, i_2, \ldots, i_t)$. The following general algorithm can be used (see [3]).

Route generation algorithm:

Step 1. If $y_{ij} = 0$ for all arcs (i, j), it is not possible to generate any route. Otherwise select an arbitrary arc (i_1, i_2) with $y_{i_1,i_2} > 0$. Set $y_{i_1,i_2} = y_{i_1,i_2} - 1$ and $t = 2$.

Step 2. Repeat while $i_1 \neq i_t$: select any arc (i_t, i_{t+1}) with $y_{i_t,i_{t+1}} > 0$. Set $y_{i_t,i_{t+1}} = y_{i_t,i_{t+1}} - 1$ and $t = t + 1$.

Theorem 3.2. If $\sum_{i=1}^{n} y_{ij} = \sum_{k=1}^{n} y_{jk}$ for all $j = 1, 2, \ldots, n$, then the Route Generation Algorithm correctly generates cycles.

Proof. If, in step 2, the path $(i_1, i_2, \ldots, i_t)$ is not closed, i. e. if $y_{i_t,j} = 0$ for every node j, then it holds that

$$1 \leq \sum_{i=1}^{n} y_{i,i_t} \neq \sum_{j=1}^{n} y_{i_t,j} = 0.$$

This contradicts the assumption. □

Comment. The Route Generation Algorithm gives us a set of routes, which depends on the choice of the initial arc in Step 1 and on the choice of the arc in Step 2. A heuristic modification of the algorithm, in which the number of reloadings of goods among different routes is minimized, is proposed in [3].

4. COMPLEXITY OF SDPDPT

Proposition 4.1. SDPDPT is NP-hard.

Proof. We reduce a decision form of the metric Travelling Salesman Problem to SDPDPT. First we recall the formulation of metric TSP.

Metric TSP problem:

- *Input:* A complete graph with n nodes and a matrix C of weights. The matrix C satisfies the triangular inequality. A number $L > 0$ is given.
- *Question:* Is there a Hamiltonian cycle of length less than or equal to L?

Define

$$Q = n - 1, \quad D = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

and solve SDPDPT. Its optimal solution is denoted $\mathcal{R} = \{R_1, R_2, \ldots, R_s\}$, where R_i is the ith cycle. Its length is denoted $h(R_i)$. The sum of all lengths is $F = \sum_i h(R_i)$. We suppose that node 1 lies on the cycle R_1. The cycles of $\mathcal{R}$ must cover all nodes since the matrix D forces us to deliver one unit from the first node to each node.

We will prove that

$$\text{a Hamiltonian cycle of length} \le L \text{ exists} \quad \longleftrightarrow \quad F \le L.$$

Moreover, we find such a cycle if it exists.

- **Case A:** $F \le L$**.** If there is a single cycle R_1 in the optimal solution $\mathcal{R}$, i.e. if $s = 1$, this cycle is Hamiltonian. Otherwise, when $s \ge 2$, we join each pair of cycles sharing a common node. We claim that there exists a cycle R_j sharing a common node with R_1. [*Proof.* If R_1 shares a common node with none of the other cycles, then we cannot transport one unit of goods from node 1, covered by R_1, to the nodes outside R_1. But the matrix D forces us to transport one unit from node 1 to all remaining nodes.] Let cycles R_1 and R_j share a common node. According to the notation of nodes in Fig. 1, the common node of cycles R_1 and R_j is the node X, the preceding node of X on R_1 is the node A_1 and the following node is A_2. The node B_1 precedes the node X and node B_2 follows the node X on the route R_j. The arc (X, A_2) lies on R_1 and the arc (B_1, X) lies on R_j. We remove those arcs and add a new arc (B_1, A_2). Then we obtain one cycle, see Fig. 2. The length of the cycle is no more than $h(R_1) + h(R_j)$ due to the triangular inequality. Joining all cycles, the cycle R_1 is Hamiltonian and its length is less than or equal to L.

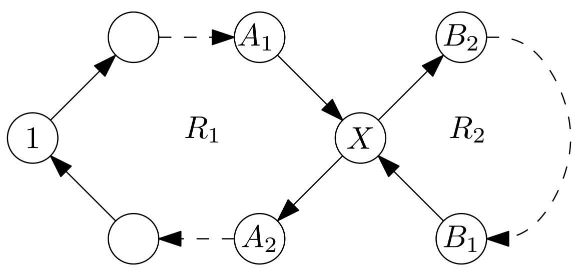

Fig. 1. Routes R_1 and R_j before catenation.

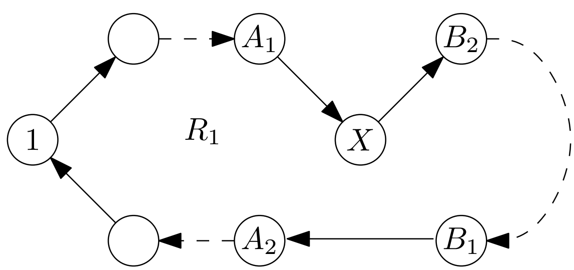

Fig. 2. Route R_1 after catenation.

- **Case B:** $F > L$. If a Hamiltonian cycle with length no more than L exists, we have a contradiction: the Hamiltonian cycle is a feasible solution to SDPDPT and the length of a feasible solution to SDPDPT is greater than L by assumption.

□

5. DEMAND REDUCIBILITY OF SDPDPT

The first question is how to define the notion of *demand reducibility* for the SDPDPT problem. The property of demand reducibility is introduced and defined for the vehicle routing problem with split demand, see [1]. The problem consists in transporting goods from a depot to customers by using vehicles of a given capacity with minimum costs. The demands of customers can be greater than the capacity of the vehicle. An instance of the vehicle routing problem is said to be *demand reducible* if there exists an optimal solution in which each customer is served by as many fully loaded depot-customer direct trips as possible. If an instance of the vehicle routing problem is demand reducible, we can reduce demands of customers by the capacity of vehicle Q as many as possible. This part of demand will be delivered by direct round trips depot-node-depot. Next we solve the problem with reduced demands of customers, i. e. with demands reduced by as many capacities of a vehicle as possible.

Now we will define demand reducibility of the SDPDPT in a similar way.

The transport between two nodes (customers) is solved in SDPDPT. The transport demand from node i to node j is denoted as d_{ij}, and from node j to node i as d_{ji}. If both d_{ij} and d_{ji} are greater then or equal to the capacity Q, then we can use as many fully loaded vehicles on the cyclic route (i, j, i) as possible. If this solution is optimal for all pairs (d_{ij}, d_{ji}), then the instance of SDPDPT is said to be *reducible.* Hence we define the reducibility in the following way:

Definition 5.1. The instance of SDPDPT is **demand reducible** if there exists an optimal solution for which it holds $y_{ij} \geq \lambda_{ij} := \min\{\lfloor \frac{d_{ij}}{Q} \rfloor, \lfloor \frac{d_{ji}}{Q} \rfloor\}$ and $y_{ji} \geq \lambda_{ij}$ for all i, j. SDPDPT is demand reducible if all instances of SDPDPT are demand reducible.

Comment. In Definition 5.1 there exists a simple cyclic route (i, j, i) for λ_{ij} fully loaded vehicles which ensure a part $\lambda_{ij}Q$ of the transport demand of d_{ij} and d_{ji}.

Proposition 5.2. SDPDPT is not demand reducible.

Proof. We will prove the proposition using the instance of Example 3.1 showing its not demand reducibility. We will show that the solution with reduced demand is not optimal. First we reduce the demands and then we find the optimal solution for the problem with reduced demands. The direct route is $1, 2, 1$, the length is 3 and this route covers transport demand $d_{12} = d_{21} = 10$. The remaining demands are covered by the optimal route $1, 3, 2, 4, 2, 3, 1$ of length 6. Total costs are $3 + 6 = 9$. The routes do not represent an optimal solution since from Example 3.1 we know that the optimal value is 8. □

6. SKIP PICKUP AND DELIVERY PROBLEM

The skip transport consists in transporting skips (big containers, trailers) from an initial location to a destination location using vehicles (tractors), see [1]. The capacity Q of vehicles is integer and limited, usually the capacity of the vehicles is one or two containers. The demand matrix D is also integer. The optimal solution has to be integer. Therefore the integrality conditions for x-variables must be added to the model (1) – (6).

Comment. Model (1) – (6) is a multi-product flow problem. The matrix of constraints is not totally unimodular, even if $Q = 1$, and hence the polytope of a relaxation of the model is not integral; see [4].

Lemma 6.1. If the nonnegative cost matrix C satisfies the triangular inequality, then the length c_{ij} of the arc (i, j) is less than or equal to the length of any path from the node i to the node j.

Proof. Let us have the path $(k_1, k_2, \dots, k_s)$, where $k_1 = i$ and $k_s = j$. Then $c_{ij} \le c_{i,k_2} + c_{k_2,j} \le c_{k_1,k_2} + c_{k_2,k_s} \le c_{k_1,k_2} + c_{k_2,k_3} + c_{k_3,k_s} \le \dots \le c_{k_1,k_2} + c_{k_2,k_3} + \dots + c_{k_{s-1},k_s}$. □

Lemma 6.2. Let (Y, X) satisfy the constraints (1) – (6). Assume that one skip from the transport requirement d_{ij} is transported along the path $P = (k_1, k_2, \dots, k_s)$, where $k_1 = i$ and $k_s = j$. Let (Y', X') be the solution obtained from (Y, X), where the transport of this skip along the path P is replaced by the transport along the arc (i, j). Then $F(Y) \ge F(Y')$.

Proof. Directly from Lemma 6.1. □

By Lemma 6.1 it holds that each demand d_{ij} has to be covered by d_{ij} vehicles going along the arc (i, j). So, the total number of vehicles y_{ij} going through the arc (i, j) should be greater than or equal to d_{ij}. It follows that we can formulate the following mathematical model of the skip pickup delivery problem (under the assumptions of Lemma 6.1):

$$\min z = \sum_{(i,j)} c_{ij} y_{ij} \quad \text{subject to}$$

$$\sum_{(i,j),\ i \ne j} y_{ij} - \sum_{(j,k),\ j \ne k} y_{jk} = 0, \qquad j = 1, 2, \dots, n; \tag{6}$$

$$y_{ij} \ge d_{ij}, \qquad i, j = 1, 2, \dots, n; \tag{7}$$

$$y_{ij} \ge 0, \qquad y_{ij} \text{ integer}; \quad i, j = 1, 2, \dots, n. \tag{8}$$

Equation (6) is a flow equation for vehicles. Inequality (7) assures the skip transport demand D.

Proposition 6.3. Let the matrix C be nonnegative satisfying the triangular inequality. Let the transport requirement matrix D be integer. Let the capacity of vehicles be equal to one. Then the model (6) – (8) is totally unimodular.

Proof. It follows from Lemma 6.2 that all skip transport requirements d_{ij} have to be transported directly along the corresponding arcs (i, j) in the optimal solution. So the solution Y has to meet the flow equation (6) to ensure the existence of a set of cyclical routes. The inequalities (7) ensure all transport requirements D. The constraint matrix of the model (6) – (8) is a node-arc matrix, which is totally unimodular. □

Corollary 6.4. Under the assumptions of Proposition 6.3, the skip pickup and delivery problem can be solved in polynomial time.

7. CONCLUSION

The pickup and delivery problem, introduced and studied in this paper, was motivated by a case study from a logistic company assuring regular transportation of piece good among regions of the Czech Republic. The problem is solved in two phases; in the first one, total transportation costs are minimized without forming routes. The result of the first phase is a number of vehicles going through arcs. In this phase, no routes are generated and no depots are determined. This is the objective of the second phase, in which routes are sought out using heuristics. Both phases represent an original tool for solving the problem, consisting of a mathematical model in the first stage and the heuristic method in the second stage.

The main difference from the pickup and delivery problems published in literature and the problem presented in the paper consists in the following points:

1. transport demand is given in the form of demand matrix of flows of goods between pairs of nodes of the communication network,
2. transport demand can be divided into many parts which are transported separately on different routes and vehicles and they can be reloaded from one route to another route,
3. route depot can be placed in any node of the route.

The main results of this paper are: the mathematical model of the problem, proof of the NP-hardness of the problem, proof of demand non-reducibility of the problem, and reformulation of skip pickup and delivery problem model as single-product minimal-cost flow problem.

ACKNOWLEDGMENT

This research was supported by GA ČR grant No. P403/12/1947 and the project F4/11/2013 founded by the Internal Grant Agency of the University of Economics, Prague.

(Received October 23, 2012)

REFERENCES

[1] C. Archetti, R. Mansini, and M. G. Speranza: Complexity and reducibility of the skip delivery problem. Transport. Sci. *39* (2005), 182–187.

[2] G. Berbeglia, J. Cordeau, L. Gribkovskaia, and G. Laporte: Static Pickup and Delivery Problems: A Classification Scheme and Survey. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.7359, 2007.

[3] J. Fábry: Optimization of routes in pickup and delivery problem. In: Mathematical Methods in Economics, University of South Bohemia České Budějovice 2010, pp. 128–131.

[4] B. Korte and J. Vygen: Combinatorial Optimization. Springer, New York 2002.

[5] M. W. P. Savelsbergh and M. Sol: The general pickup and delivery problem. Transport. Res. *29* (1995), 17–29.

[6] S. R. Thangiah, A. Fergany and A. Salman: Real-time split-delivery and delivery time window problems with transfers. Central Europ. J. Oper. Res. *15* (2007), 4, 329–349.

Jan Pelikán, University of Economics Prague, W. Churchill sq. 4, 130 67 Praha 3. Czech Republic.

e-mail: pelikan@vse.cz